UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/717,676	11/21/2000	Bradley J. Bartz	777.346US1	9183

41505 7590 12/02/2004

WOODCOCK WASHBURN LLP
ONE LIBERTY PLACE - 46TH FLOOR
PHILADELPHIA, PA 19103

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2124	

DATE MAILED: 12/02/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/717,676	BARTZ ET AL.	
	Examiner	Art Unit	
	Tuan A Vu	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 August 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-4,6-22,24 and 26-38* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-4,6-22,24 and 26-38* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 8/10/2004.

As indicated in Applicant's response, claims 1,3-4, 6, 11, 22, 24, 27 have been amended and claims 5, 23, 25 canceled. Claims 1-4, 6-22,24 and 26-38 are pending in the office action.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. Claims 20 and 27 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

The Federal Circuit has recently applied the practical application test in determining whether the claimed subject matter is statutory under 35 U.S.C. § 101. The practical application test requires that a " useful, concrete, and tangible result" be accomplished. An "abstract idea" when practically applied is eligible for a patent. As a consequence, an invention, which is eligible for patenting under 35 U.S.C. § 101, is in the "useful arts" when it is a machine, manufacture, process or composition of matter, which produces a concrete, tangible, and useful result. The test for practical application is thus to determine whether the claimed invention produces a "useful, concrete and tangible result".

Claim 20 recites a versioned output document for developing software, the document containing changes from one previous version but excluding changes from another version of the document. These are structural elements as to the content of the versioned document but nowhere is there any functional description showing interrelationships between any of the descriptive structural elements so as to enable the realization of an action leading to a concrete, tangible, and useful result. Claiming that a document contains this feature and not another feature does not amount to any action being performed in order to yield the result as required by the practical application test as mentioned above. The claim therefore is no more than an impractical and/or abstract idea; and is rejected for leading to a non-statutory subject matter.

Claim 27 recites an association file comprising entries to be incorporated in a change set and at least one entry of a non-versioned document to be incorporated in the change set. Like the case in claim 20, the recital of a file with descriptive structural elements, e.g. entries to be incorporated in a change set, does not amount to any action depicting the interacting between those structural elements; and above, this claim does not lead to a concrete, tangible and useful result. The claim therefore is no merely an abstract idea; and is rejected for leading to a non-statutory subject matter.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. Claims 3 and 8 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 3 recites the limitation "in both of the documents" in line 4 of claim. There is insufficient antecedent basis for the limitation 'both' in the claim. Examiner will interpret this as 'in two of the documents'

Claim 8 also recites "in both of the documents" in line 4 of claim. Examiner will interpret this as 'in two of the documents'.

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

7. Claims 11-12, 15-16, and 18-21 are rejected under 35 U.S.C. 102(b) as being anticipated by Leblang et al., USPN: 5,649,200 (hereinafter Leblang).

As per claim 11, Leblang discloses a method useful in developing software having multiple versioned documents, comprising:

comparing 2 child documents of a common parent document with each other and indicating any differences as actual conflicts between the two (e.g. Fig. 3);

comparing both child documents with a common parent for indicating possible conflicts between child documents for portions therein that are same to each other (e.g. Fig. 13, 16 – Note: X being at root of tree reads on common parent to child documents in the process for detecting possible conflicts or delta);

producing a merged output document indicating both actual and possible conflicts (e.g *result of* ... merge – Fig. 13, 16).

As per claim 12, Leblang discloses delta as being recorded from previous version of child documents and being reused for unmerging (e.g. *substractive merge* - Fig. 16 – Note: keeping track of previous conflicts in order to perform selective unmerging reads on alternative histories of child documents being stored).

As per claim 15, this is a computer-readable medium version of claim 11. As for a medium to support the software product, Leblang(Fig. 1, 2, 6) discloses use of workstations to develop the software building, hence implicitly discloses the use of computer-readable medium.

As per claim 16, Leblang discloses a method useful in developing software having multiple versioned documents, comprising:

receiving first, second, and third version levels, where the third version incorporates 2nd set of changes from a second version, and the second version incorporates 1st set of changes from the first version level (e.g. Fig. 13, 16); unmerging the first set of changes from one document at the 3rd version level, while preserving the 2nd set of changes (e.g. *substractive merge* – Note: substract Δa while maintaining Δb and Δc reads on preserving 2nd level and unmerging the first set)

outputting of merged documents (e.g. result of ... merge – Fig. 13, 16).

As per claim 18, Leblang discloses a developer receiving versioned objects for modification and revision checking hence has implicitly disclosed receiving indications of 1st, 2nd, 3rd version levels (e.g. col. 23, line 61 to col. 24, line 62).

As per claim 19, Leblang discloses a computer-readable medium version of claim 16 (e.g. Fig. 1; col. 5, lines 32-53).

As per claim 20, Leblang discloses a versioned output document containing changes from one previous version of the document, but excluding changes from another previous version of the same document (e.g. Fig. 16 – Note: substract Δa while maintaining Δb and Δc reads on containing changes from one previous version of the document, but excluding changes from another previous version)

As per claim 21, Leblang discloses other previous version of the document is a previous version of one document (e.g. X + Δa; X + Δa + Δb – Note: X is the one document with previous version stored in level of hierarchy)

8. Claims 22, 24, and 26-29 are rejected under 35 U.S.C. 102(b) as being anticipated by Carrier III et al., USPN: 5,903,897 (hereinafter Carrier).

As per claim 22, Carrier discloses a method useful in developing software having multiple versioned documents, comprising:

associating like versions of versioned documents with each other in a change set specification (e.g. *released forms* – col. 6, lines 17-52; Fig. 5-6 – Note: like versions reads on file selected for a release) by listing each of the versioned documents in a association file that designates the like versions to be incorporated in the change set specification (e.g. *source modules* step 160, *release forms* step 162, Fig. 3; ; col. 4, lines 54-62 --Note: release from different reasons reads on change set and multi association of source modules checked in a release form reads on listing of versioned documents in a association file)

associating additional, non-versioned documents into the same change set (e.g. step 328 – Fig. 9; *build log 550, test cases 520, build report 552*– Fig. 12);

retrieving both versioned and nonversioned documents as a single unit (e.g. *source files and check-in info, defect tracker* -Fig. 11; Fig 12);

storing the association file in a memory separately from the documents listed in the file (e.g. Fig. 1; *check-in data* – col. 3, line 62 to col. 4, line 10; col. 4, line 48 to col. 5, line18 - Note: source files being in database of software control system and later attached to a logical grouping called *release forms* being prompted at the release building tool reads on association file residing --in the load builder-- separate from source files being persisted and checked in a version database)

As per claim 24, Carrier discloses associating nonversioned and versioned documents by listing them in a same association file (e.g. e.g. *release forms* → *build 170* – Fig. 3; *build report 552* – Fig. 12).

As per claim 26, this is a computer medium version of claim 22 with the medium limitation being disclosed by Carrier (medium driver *60* – Fig. 1).

As per claim 27, Carrier discloses an association file (*released forms* – col. 6, lines 17-52; Fig. 5-6) for a set of versioned documents, comprising

a plurality of entries each designating a version of the versioned documents to be incorporated in a change set (e.g. *source modules* step 160, *release forms* step 162, Fig. 3; col. 4, lines 54-62 --Note: release from different reasons reads on change set);

at least one entry designating a non-versioned document pertaining to at least one versioned documents to be incorporated in a change set (e.g. step *328* – Fig. 9; *build log 550, test cases 520, build report 552*– Fig. 12).

As per claim 28 and 29, Carrier discloses incorporating in the change set non-versioned design documentation (e.g. col. 8, lines 20-36) and bug report (e.g. reporting tool 24, defect tracker 32, build report 52 – Fig. 1; *release form ... report number* - col. 4, line 54-62).

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.
>
> Note: 35 U.S.C. § 102(e), as revised by the AIPA and H.R. 2215, applies to all qualifying references, except when the reference is a U.S. patent resulting directly or indirectly from an international application filed before November 29, 2000. For such patents, the prior art date is determined under 35 U.S.C. § 102(e) as it existed prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. § 102(e)).

10. Claims 31-38 are rejected under 35 U.S.C. 102(e) as being anticipated by Hopwood et al., USPN: 6,223,343 (hereinafter Hopwood).

As per claim 31, Hopwood discloses a method useful in developing software having multiple versioned documents, comprising:

synchronizing a set of files from a common storage area to a private enlistment area (e.g. e.g. col. 19, lines 36-63; Fig. 3; col. 17, line 50 to col. 18, line 39; *workstation 96* – Fig. 7 – Note: *host developer workstation* reads on enlistment area where files in RMS are checked out for synchronization and selection by developer);

adding a set of build-specific changes to the files in the enlistment area (e.g. col. 17, line 50 to col. 18, line 39);

making local changes to the enlisted files (e.g. col. 19, lines 36-63; Fig. 3; step *S50* – Fig. 8);

thereafter removing the changes from the enlisted files and returning enlistment files to the common area (e.g. *master copy* - col. 18, lines 40-52 – Note: master copy is equivalent to storing/keeping of copies without added changes back into the repository RMS).

As per claim 32, Hopwood discloses repeating the enlistment process (e.g. *inventory groups* – Fig. 8 – Note: plurality of inventory groups reads on repeating of enlistment process).

As per claim 33, Hopwood discloses an common area for all separate enlistment areas (e.g. col. 15, line 4-35 – Note: per host workstation, a RMS is a shared area for all enlistments viewed by the user of that workstation, hence common to all separate views).

As per claim 34, see Hopwood, Fig. 7; *synchronize inventory, merge elements* -Fig 12; refer to *enlistment area* of claim 1.

As per claim 35, Hopwood discloses merging with a set of previous local changes (e.g. col. 19, lines 36-63; *synchronize inventory, merge elements* -Fig 12).

As per claim 36, Hopwood discloses getting the build-specific changes from a build area (e.g. Fig. 13-14; col. 17, line 50 to col. 18, line 39 – Note: locked by issuance reads on getting build-specific changes); and merging the build-specific changes into the enlistment files (e.g. see claim 35 – Note: merging different files from RMS – see Fig. 6-7, 13 – 14 -- in order to group into a set for a build/issuance reads n merging build-specific changes).

As per claim 37, Hopwood discloses selecting a particular build from which to get the specific changes (Note: the involvement of developer and an issuance instance from Fig. 6-7, 13-14 implicitly disclose selecting of a particular build).

As per claim 38, this is a computer-readable medium version of claim 1. As for a medium to support the software product, Hopwood discloses use of workstations to develop the software building (e.g. Fig. 7) hence implicitly discloses the use of computer-readable medium.

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

12. Claims 1-2 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hopwood et al., USPN: 6,223,343, in view of Leblang et al., USPN: 5,649,200.

As per claim 1, Hopwood discloses a method for developing software having multiple versioned documents, comprising:

comparing multiple ones of the versioned elements of a project at different subdivision level (e.g. *merge elements 16* – Fig. 12; Fig. 15-20; Fig. 8, 11; col. 14, lines 32-36; col. 18, lines 23-46; *merge* -col. 19, lines 36-47 – Note: management of change to level of logical group elements to record changes using merge tool is equivalent to comparing versioned) and indicating the changes on elements of workgroups at each subdivision level (e.g. *record of ... changes* - col. 19, line 47 to col. 20, line 10; Fig. 15);

unmerging from a later version of versioned documents of changes previous to a further set of changes (e.g. *regression, restore, revert* – col. 28, lines 45-55);

associating with the set of changes in versioned documents a plurality of non-versioned documents pertaining respectively to versions of versioned documents (e.g. *issuance control data, maintenance libraries, audit trails, issuance reports, metadata* - col. 13, line 63 to col. 14, line 14; *difference reports* - col. 19, lines 36-40 – Note: each element retrieved for work group build is equivalent to versioned document);

updating by copying files from a common storage area to a private enlistment area, adding build-specific changes to enlistment copies, making changes to the modified copies (e.g. col. 19, lines 36-63; Fig. 3; *host developer workstation* - col. 17, line 50 to col. 18, line 39; *workstation 96* – Fig. 7) and thereafter removing the changes and returning the files to the common area (e.g. *master copy* - col. 18, lines 40-52 – Note: master copy is equivalent to storing/keeping of copies without added changes back into the repository).

But Hopwood does not explicitly specify comparing of versioned documents to a common parent document and indicating conflicts caused by alternative histories from the parent document. Hopwood teaches merging and validating of software elements persisted in a

hierarchy of a version control RMS (e.g. Fig. 11; col. 18, line 40 to col. 19, line 48; col. 15, lines 14-26; Fig. 8). which suggests conflict resolving with changes in the hierarchy of version tree and comparing modified file against persisted earlier change. Leblang, in a method to compare/merge files in a hierarchy of development environments similar to built environment as taught by Hopwood, discloses comparing multiple versioned child documents to a parent document and recording conflicts or deltas being reused for selectively removing delta (e.g. Fig. 13, 16 – Note: highest level base level of tree is parent). It would have been obvious for one of ordinary skill in the art at the time the invention was made to add to Hopwood's method of merging software elements and auditing changes to child-parent comparison technique as taught by Leblang, in case Hopwood's RMS comparing method does not already include one, because this would enable the synchronizing child version from its ancestor version with incremental difference extracting and identification of conflicts by means of tree-like branching, thus enabling better reconciliation of hierarchical versioned documents and possible removing of selective delta as taught by Leblang.

Nor does Hopwood explicitly specify comparing versioned documents at a plurality of different subdivision levels. In view of the teachings by Hopwood's RMS file difference resolving and Leblang's effect merge/unmerge at different levels of project/documents hierarchy, this limitation is implicitly disclosed in the combination of Hopwood/Leblang from above.

Nor does Hopwood disclose preserving the further set of changes when unmerging from a later version. But in view of Leblang's teachings (*substractive merge* – Note: substract Δa while maintaining Δb and Δc reads on preserving 2nd level and unmerging the first set), it would have been obvious for one of ordinary skill in the art at the time the invention was made to

provide such delta recording and selective unmerging by Leblang because this would enable a bi-directional merging and generating on demand of more version combinations such to obviate the otherwise burden of systematically restarting the incorporation of deltas from a parent from the top (see Leblang col. 25, lines 1-29).

As per claim 2, this is a computer-readable medium version of claim 1. As for a medium to support the software product, Hopwood discloses use of workstations to develop the software building (e.g. Fig. 7) hence implicitly discloses the use of computer-readable medium.

13. Claims 3-4, and 6-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Skinner, USPN: 5,481,722 (hereinafter Skinner) in view of Panagiotis, *Diff* Man Page/Man-cgi 1.11, 1994 (hereinafter Panagiotis)

As per claim 3, Skinner discloses a method useful in developing software having multiple versioned documents, comprising:

defining subdivisions in 2 documents to be compared (e.g. *start control line*, *end control line* – Fig. 6; *text line* – Fig. 6 Note: grouping of lines under limits of control lines reads on first division being a section of document; and subdivision reads on lines enclosed in each first divisions);

comparing current subdivision of one documents to a current subdivision of other document; and indicating differences therebetween (e.g. *output table* , *branch deltas* – Fig 7a-b; *control line* - col. 7, line 30 to col. 8, line 25);

comparing current subdivision or lines of one documents to a current line of the other document (Fig. 6 – Note: the comparing of lines by lines text is implicitly disclosed because delta text lines 240 within each block delimited by control line reads on line division being

compared to be part of a delta result); repeating second comparing step within this subdivision type (e.g. *text lines* - Fig. 6);

repeating first comparing step for further lines within the documents (Fig. 6; *control line* - col. 7, line 30 to col. 8, line 25 – Note: delta text lines 240 within each block delimited by control line, reads on line division being compared to be part of a delta result);

producing an output document indicating differences found in both comparing steps (e.g. *output delta table* - col. 8, lines 16-25; Fig. 5).

But Skinner does not explicitly disclose defining subdivisions within a line of the 2 documents and comparing a current such subdivision of one document with a current subdivision of the other document to indicate differences; and repeating such comparing step for further subdivision within the current line of the 2 documents. A line is composed of blank space or characters, something inherent to a line. The comparing of line necessarily encompasses the going through of its elements; and such is implicitly disclosed by Skinner. In case Skinner does not already provide subdividing a line into a character division to perform character comparison with a counterpart in the current line of the 2 documents, there is suggestion as to why Skinner should have done so. Skinner is using an Unix platform by Sun Microsystems, and suggests the use of command line utilities within such platform (see Background and col. 6, lines 10-32). In a man page for *diff* similar to Unix Man page, Panagiotis discloses line by line comparison with options –a or –c or -# and option i to ignore character and –b to address blanks; thus has strongly indicated partition a document into line and character subdivision while scanning a line. Thus , it would have been obvious for one skill in the art to provide the Unix utilities as taught by Panagiotis to Skinner Sun Microsystem computer facilities to that the subdivision comparing as

taught by Panagiotis is utilized because this would enable Skinner to make use of utilities already available in its platform and thereby be endowed with options as suggested above to selective address line-by-line pattern and its character subdivisions and make the document comparing process more time and cost efficient.

As per claim 4, as suggested by Skinner, each control line is indicative of a revision/edition level so such control line entails a matched version (e.g. *unchanged, may be zero* -- col. 7, lines 37-53). Skinner has implicitly disclosed that upon detecting no difference between first-type subdivision, the comparing of second-type subdivision is inhibited. The limitation as to compare line by line and detecting no difference has been addressed in claim 3 using the combination Skinner/Panagiotis.

As per claim 6, Skinner discloses that the first subdivision type is a line(*start control line*, *end control line* – Fig. 6; *text line* – Fig. 6). But Skinner does not explicitly teach that the second subdivision is a character but since a character matching is inherent to a line comparison, and moreover should Skinner not already implicitly disclose this, that subdivision being a character has been would have been obvious in view of the rationale set forth in claim 3.

As per claim 7, this is a computer-readable medium version of claim 3. As for a medium to support the software product, Skinner (Fig. 3) discloses use of workstations to develop the software building, hence implicitly discloses the use of computer-readable medium.

As per claim 8, Skinner discloses a method useful in developing software having multiple versioned documents, comprising:

defining at least 2 nested types of subdivision in both documents (e.g. *start control line*, *end control line* – Fig. 6; *text line* – Fig. 6 – Note: second subdivision type is the line-by-line type in a group delimited by control words);

comparing current first-type subdivision of one documents to a counterpart of other document; and indicating differences therebetween (e.g. *output table , branch deltas* – Fig 7a-b; *control line* - col. 7, line 30 to col. 8, line 25 – Note: difference in blocks reads on first type subdivision comparing);

for a current second type division, comparing current second-type subdivision of one documents to a counterpart of the other document and indicating differences therebetween (Fig. 6; *delta text lines 240* -col. 7, line 30 to col. 8, line 25 – Note: the comparing of lines by lines within a block text is implicitly disclosed);

repeating second comparing step within subdivision of second type (e.g. *text lines* between *control lines* -- Fig. 6);

repeating first comparing step for further first-type subdivision within the documents(e.g. *control lines* per block - Fig. 6; col. 7, line 30 to col. 8, line 25);

producing an output document indicating differences found in all comparing steps (e.g. *output delta table* - col. 8, lines 16-25; Fig. 5).

But Skinner does not explicitly disclose defining 3 nested types of subdivision in documents; and for each subdivision, comparing a current level subdivision of one document with a current subdivision of the other document; and repeating the comparing at each level such as 3rd level subdivision, 2nd subdivision comparing and 1st subdivision comparing against respective counterparts in both documents. But in view of the teaching by Panagiotis and

Skinner's method being implemented in an Unix facilities, the subdividing of document into blocks of strings by Skinner (1st type) and line subdivision (2nd type) then character subdivision (3rd type) as by Panagiotis such that 3 nested subdivision are defined for text file comparison and iterative comparing each level of subdivision at a time would have been obvious according to the rationale as set forth in claim 3;

As per claim 9, see claim 4 for corresponding rejection.

As per claim 10, Skinner discloses a multi-line section enclosed by control lines (e.g. Fig. 6, i.e. multi-lines section).

14. Claim 17 is rejected under 35 U.S.C. 103(a) as being unpatentable over Leblang et al., USPN: 5,649,200.

As per claim 17, Leblang does not specify that the unmerged result being generated (see Fig. 16) as being a 4th level version. But based on the version control stored in hierarchy of version tree (see col. 25, line 11 to col. 26, line 18; Fig. 13), the suggestion of another level different from the 1st three level of Fig. 16 is strongly suggested and this was a known concept in the methodology of hierarchizing version of files in most software maintenance at the time the invention was made. Hence, it would have been obvious for one of ordinary skill in the art at the time the invention was made to create the unmerged version a file located at a fourth level different from the other 3 levels because for each file with a specific set of change, a specific version should be attached to and such version should belong at one level of the tree as mentioned by Leblang so no 2 non-identical documents stay in one version level.

15. Claims 13-14 are rejected under 35 USC 103(a) as being unpatentable over Leblang et al., USPN: 5,649,200, as applied to claim 11, in view of Howard, USPN: 5,600,834 (hereinafter Howard).

As per claims 13 and 14, Leblang discloses using delta recording as a result of comparing in the merge and indication conflicts due to alternative histories of child documents (Fig. 16) but fails to disclose marking of possible conflicts (re claim 13) or actual conflicts (re claim 14) in the merged document. Howard, in a method to reconcile versions of a file with generation of a merge document analogous to the combined merge by Leblang, disclose a document combining the results from reconciling documents with history of more than one versions (e.g. Fig. 4; col.6, lines 35-48) and marking in this document of possible conflicts due to histories of child/parent documents (e.g. col. 12, lines 11-12). It would have been obvious for one of ordinary skill in the art at the time the invention was made to provide such information marking as suggested by Howard within the delta combination file as taught by Leblang because it would provide additional information for the developer or version administrator to further effect verification or error-proofing on the combined merge output file prior to committing it to persistent storage or distribution to sites as suggested by Howard.

16. Claim 30 is rejected under 35 U.S.C. 103(a) as being unpatentable over Carrier III et al., USPN: 5,903,897.

As per claim 30, Carrier does not disclose including screen shots in the non-versioned documents although Carrier discloses problem report to associate with a release (re claim 29). Official notice is taken that the use of screen/snap shot to take instant capture of an malfunction message or error display during a debug for report on a software execution or product assessment

such as suggested by Carrier was a well-known practice at the time of the invention. Hence, it would have been obvious for one of ordinary skill in the art at the time the invention was made to include in the association file as taught by Carrier any debug information, e.g. including screen shots as taught by known practices because this would provide graphical evidence for the developer to better address conflicts or bugs thereby ensure quality in delivering versioned elements for the build. The arguments therefore are not persuasive.

Response to Arguments

17. Applicant's arguments filed 8/10/2004 have been fully considered. Most arguments are becoming moot in regard to the new grounds of rejection. With respect to some arguments following are the Examiner's observation and response thereto.

Applicants have submitted that Carrier does not teach 'storing the association file in a memory separately from the documents listed in the file' (Appl. Rmrks, pg. 13, middle para). The rejection is not pointing to the fact the since the release forms for logically enlisting the versioned files being in the check-in storage, the release forms are considered the association file being prompted at the developer building tool interface and completed dynamically in such environment; and this is a separate area from a database (see VC databases 12 of Fig. 1) where versioned documents are persisted as opposed to dynamic form being prompted. Because there is no disclosing that such database also stores versioned release forms, it is therefore reasonable to read it as though the release forms are not stored in the same repository of the versioned files to be loaded in the builder area.

Conclusion

18. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)272-3719.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence – please consult Examiner before using) or 703-872-9306 (for official correspondence) or redirected to customer service at 571-272-3609.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VAT
November 29, 2004

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100